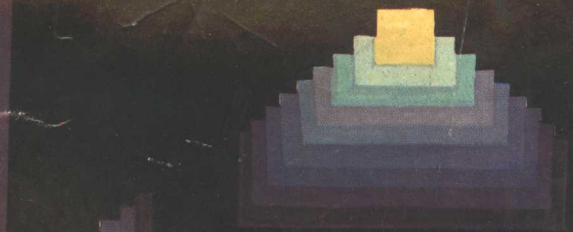
数据结构技术



山东科学技术出版社

数据结构技术

〔美〕托马斯 A·斯坦达斯 著
王雨生 宋伶伶 等译编
宋 健 赵珀璋 主 审

山东科学技术出版社
一九八七年·济南

译编者还有：王　宏　宋　俐　朱建平　孙慧敏
责任编辑：孟爱平

数据结构技术

〔美〕托马斯·斯坦达斯 著
王甫生　宋伶伶　等译编
宋　健　赵珀璋　主　审

*

山东科学技术出版社出版
（济南市玉函路）
山东省新华书店发行
山东新华印刷厂印刷

*

850×1168毫米32开本　15印张　4插页　347千字
1987年12月第1版　1987年12月第1次印刷
印数：1—3400
ISBN7—5331—0176—6/TP·1
书号 13195·180　（软精装）定价 4.70 元

前　言

随着电子计算机科学的飞速发展，计算机已在各行业中得到普遍应用。计算机应用的一个重要方面，是数据处理和实时控制，如情报检索、数据采集、企业管理、档案管理、预测决策分析、人工智能等。将计算机应用到这些领域中所面临的第一个问题，就是信息量大，种类多，结构复杂。因此，必须设计新的更高级的数据结构和组织结构，以便有效地实现数据采集、数据组织、数据存储、数据传输、数据处理和数据利用。

数据结构研究数据的组织形式，分为抽象数据结构和内部存储结构。抽象数据结构包括数据串、数组、队列、堆栈、表、树和有向图；内部存储结构包括向量、链表和丛等。

数据结构是编译构造、操作系统、数据库、信息检索和人工智能等课程的基础。它作为一门课程，最早是由美国人在1968年提出的，当时对这门课程的范围还没有明确的规定。十几年来，数据结构技术有了很大的提高，已成为计算机专业必修的基础课。自1981年起，我国各高等院校也陆续开设了这门课程。

本书是为学习计算机的大学生和计算机工作者、专业编程人员编写的。主要论述存储器分配和管理的数据结构、相应的算法知识、程序设计技术，分析在处理各种各样的数据结构中建立起来的方法。本书重点分析了列表、树、字符串、字节、堆栈、文件、表及多链结构的处理方法，详细解释了散列编码、搜索、紧凑、无用单元收集、加密、转换、压缩等重要过程。

编写本书的目的，一是给专业程序设计人员和计算机科学

工作者提供高水平的专业知识，二是作数据结构课程的基础教材。基于这两个目的，一方面介绍、综述并比较了计算机领域中包括最新发展成果在内的大量的重要资料，提供了许多数据结构文献的参考书目；另一方面，注意提供适合不同水平和不同学时的学习内容，读者可根据自己的情况酌情选读。

初级课程

章	节
1	§1·1，§1·2，§1·3
2	§2·1，§2·2，§2·3，2·4
3	§3·1，§3·2，§3·3(一)，§3·4(一、二)，§3·5，§3·6(一、二)，§3·7
4	§4·1，§4·2(二)，§4·3(一、二、三、十)，§4·4，§4·5
5	§5·1，§5·2，§5·3，§5·4(一、二)，§5·5
6	§6·1，§6·2(一)，§6·3
7	§7·1，§7·2，§7·4，§7·5(一)
8	§8·1，§8·2，§8·3(一)，§8·4
9	§9·1，§9·2，§9·3(一、二、三、五、六)

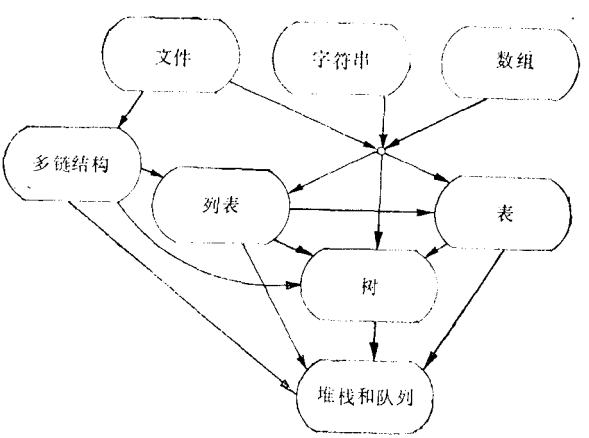
高级课程

章	节
1	§1·1
2	§2·1，§2·2，§2·5
3	§3·1，§3·3，§3·4(一、二)，§3·5(一)，§3·6(二)，§7·3

章	节
4	§4·1，§4·2，§4·3
5	§5·1，§5·3(三)，§5·4，§5·6，§5·7
6	§6·1，§6·2，§6·3
7	§7·1，§7·2，§7·2(六)，§7·3，§7·5(二)
8	§8·1，§8·5
9	§9·1，§9·2，§9·3

本书以数据结构为主，在深度和广度上包括了我国现行教学大纲的全部内容，还包括了算法分析的主要内容。本书与一般关于算法的专门书籍的区别在于：把数据分类方法看作不同数据结构的图解，只作简单阐述。在关于堆栈和队列的章节中讨论快速分类，在关于树的章节中讨论堆分类。本书比算法分析书提供了更多关于各种初级数据表达技术方面的知识，但并不极力为算法论题中的搜索、数据分类等提供比较算法分析，在这个意义上强调和证明了数据结构的性能特性。

本书的内容编排顺序与其它数据结构书明显不同。它是在



通过分析各章节的内在联系后，得出如上图所示的结果。从中可以看出，框图中不再含有任何循环，这意味着存在着一种顺序，这种顺序排除了各章节间互为前提又互相依赖的弊病。书中的每个概念的应用都出现在这个概念定义之后。

本书在编写过程中，得到了我国著名控制论、系统工程专家宋健教授和计算机专家赵珀璋高级工程师的大力支持，并主审了全书。还得到了于景元教授、王义福教授、陈玮研究员的关怀和支持。黄伟敏、王鲁彬也为本书作了不少工作。在此，一并表示衷心感谢。

由于我们水平和时间所限，书中难免出现错误，敬请读者正之。

编　者

1987年2月

目　录

第一章 绪 论

§1·1 引 言

数字计算机是一种特殊的解决问题的装置。它的多功能性，相当程度上取决于程序设计能力，即在计算机中有效地表达和处理大量信息而编制程序的能力。

在裸机上所能提供的基本信息结构，通常是简单的、无弹性的和序列排列的。这种基础通用信息存储介质必须形成更高级的结构，这个结构能够忠实地再现解决问题所需要的信息，并能保存应用于分步解决问题而进行运算所必需的重要信息。

为了将机器级信息结构编制成多功能和高级的数据描述，现在已有了庞大的技术体系。例如，有很多有效地用于存储器分配、动态结构变化控制、具有所需特性的数据项搜索和完成大量其它工作的算法。

对于这些数据表达技术和有关处理算法的加深，是计算机科学的核心。

研究数据结构的关键在于数据表达的概念。粗略地说，我们可以把所给的表达介质的数据和处理方法加以组织，建立数据表达式，使其满足该式所代表的某个特定问题和操作所需要的性能和特征。关键的问题在于对所需行为的真实模拟。在某些简单情况下，它可用现代逻辑、一公理化系统模型的概念形式地导出。在这个系统中，公理能规定问题和运算的特点要求，而模型能提供对客观事物的精确的模拟。

通常有两种不同的语言级与表达过程相联系。一个用于描绘所模拟的结构，另一个则在基础介质中描述问题的目的和操作。在某些系统中，要用两种以上的语言级，即在高层和低层的表达介质之间需要有一个中间的层次。

这样，数据表示方法在机器域和问题域之间常常要有一次或多次的分级。例如，在一个民航定票系统里，问题域的实体级上可能包括调度、飞行计划、日期和预定民航客票。在系统实现时，中间级的实体可能包括文件、表格、记录、串等，而在机器级实体中可能包括用机器字表示的字节和线性序列等。

在这本书中，我们将主要讨论通用的、中间级的数据结构（如文件、列表、数组和信息串等）和它们在机器级上的基本数据表达方式。经验表明，这些通用的中间级数据结构在应用程序中有很高的实用价值。一个给定的中间级结构（如一个矩形数组或数表）通常要有最基本的表达范围，在每一个范围内都有不同的时间和空间的资源要求。

本书的主要目的在于扩展对于各类中间级数据结构范围的表达技术的理解。训练读者并使他们能在其中进行适当的选择，或设计新的数据结构以适用于新的目的。

§1·2 基本概念和符号

一、单元、字段和指针

数据的表达式经常存储在存储介质之中，后者又由分离的可编址单位所组成。可编址单位的地址常常由指定范围（$m \leqslant i \leqslant n$）内的整数组成。例如，含有262144个字长为36位的字的存储器中，字的地址可以由在区间（$0 < i < 2^{18} - 1$）中的整数组成。这样的地址可以由18位二进制整数表示。

一个单元是一个可编址的存储单位，而单元又可分为命名字段。例如，图1·1中，地址为α的单元具有两个字段，分别命名为信息字段(info)和连接字段(Link)。

图 1·1

在本书中，α和β经常用来表示单元的地址。后面跟有冒号的单元地址通常放在单元的左侧。单元字段的名字通常附加在单元图的边上（通常在上面，有时也可能在图的两边或下方）。在图1·1中，x_1是信息字段的内容，β是连接字段的内容。一般说来，字段是数据序列指定的部分，通常由相邻符号的子序列组成。本书假定：一个单元的字段之间没有重迭，但不能假定单元的这些字段与可编址存储单位完全一致，也不能假定这些字段可单独编址。

我们将始终假定，不同的单元有不同的地址。单元的地址都不必与基础存储介质中的不同可编址单位准确对应。单元有时可以是一组相邻的基本可编址单位。例如，在一个存储器中，它的可编址单位的字长为36位。可以规定，这个存储器可以划分为单元，而每个单元由相邻的两个字组成。在这种情况下，每个双字单元的地址采用它的第一个字的地址（即最低编址）。习惯上，多字单元的地址是用它的第一个字的地址。这样，单元就是我们施加于基本存储介质的一种组织形式。在这种介质里，把它的可编址单位分为成对的、不相交的可编址单位的相邻组。同时，取可编址单位最低字的地址作为单元的地址。有时也把多字单元叫做块。

当单元A的字段F包括另一个单元β的地址时，就说A的F字段包括着一个指向β的指示字（等于说A的F字段与β有联系）。图1·2所示的两个单元分别有一个包含着指向另一个单元的指示字的连接字段。

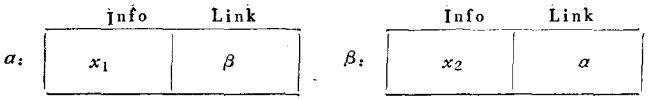


图 1·2

一般用箭头来表示指示字的出现。例如，图 1·3 给出了与图 1·2 完全相同的信息，每个箭头的尾巴都驻在存储着指示字地址的连接字段里，而每个箭头都指向指示字所指向的地址。

在画图时，经常采用连接单元，这时，有些单元的地址并不明显地标出。图1·4所示的就是这样的情况。

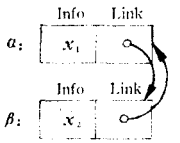


图 1·3

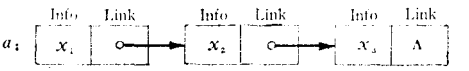


图 1·4

在图 1·4 中，尽管只有最左边的单元标出了明确的地址，其它单元的地址无论是否标出，都隐含着其它各个单元都具有不同的地址。图 1·4中出现的箭头，每个都对应着一个不同指示字的地址。此图还说明了一种把单元连接成链的技术。这里，除了最后一个单元外，每个单元都指向下一个单元。这种连接起来的单元叫做单元连接表。请注意，图 1·4 中最右边的单元连接字段包含着符号Λ。在本书中，Λ表示一个专用的零位指示字，表明这个连接表的终止。我们假定，它是区别于所有其它单元地址的地址。

二、变量与赋值

在文字描述和算法中，我们将使用变量一词(如x或Head)。变量有各种数量值，通过赋值语句，把值赋给变量。例如，执行赋值语句$x\leftarrow 3$，结果使变量x的值成为整数 3 。执行赋值语句Head$\leftarrow\alpha$，使得变量Head的值成为地址α。在任何时候，变

量的值都等于最后赋给它的值。赋值语句Head←a，用新值a代替了Head的前一个值。直到使用新的赋值语句，把不同于a的值赋给Head之前。Head一直保留a这个值。当变量以一个单元的地址作为它的值时。可通过从一个变量指向该单元的指针，表示它的地址值。例如，假定一些单元的集合连接在一起，并且已完成了赋值语句Head←a，那么，就能得到图1·5所示的结果。

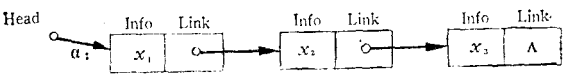


图 1·5

符号Info(Head)指定单元信息字段的内容，该单元的地址就是变量Head的值。这样，在图1·5中，Info(Head)=x_1。类似地，Link(Head)是图1·5中的中间单元的地址，而且，Info〔Link(Head)〕=x_2。把Info(Head)放在赋值语句的左边，Info(Head)←x_4时，就指定了单元信息字段的值由新值x_4代替，这个单元则由Head指定。例如，在图1·5中，开始执行赋值语句Info(Head)←x，便得出一个新图（图1·6）。

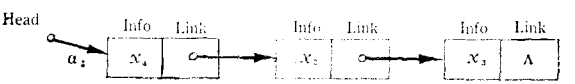


图 1·6

单元最左边的信息字段已变为新值x_4。同样，执行三个赋值语句时，Temp←Link (Head)，Link (Head)←Link〔Link (Head)〕，Link(Temp)←Λ，便使图1·6变为图1·7。

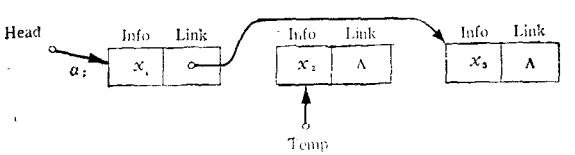


图 1·7

三、算法的符号

本书中用于表示算法的标志符号，与D.E.Knuth 的关于计算机程序设计艺术的一系列著作（见Knuth〔1973a〕）中所采用的标志符号类似，也与广泛应用于数据结构方面的文献中的标志符号基本上一致，有时只有微小的差别。为了说明本书中所采用的标志符的差异，下面考查一个查表算法的例子。

在数据结构应用中，通常首先记录，然后检索与符号项有关的信息。例如，在编译一个FORTRAN程序时，常需要在编好的程序中建立FORTRAN标志符与它的存储单元之间的对应关系。为此，在编译程序中常常使用符号表。这样的符号表可以对编译程序中的每个不同的标志符I记录一个地址α。每当出现一个标志符I的具体值时，就可以去查这样的表，以确定I是否在表中，若在，那么它以前与什么地址α相连。执行这个任务的一个方法是仅在表中存一个标志符目录，每次查找都要通查全部目录。这样一来，如果有许多标志符的话，就太费时间了。如果采用标志符I的表达式，作为在表中的直接索引，查找时就容易得多了。但是，由于FORTRAN的标志符是由 1 ～ 6 个字母组成，或者由字符型的数字组成，所以就可能有1617038306个不同的FORTRAN标志符。显然，在任何实际编译程序中，象这样的直接索引表将会大到无法使用的程度，所以转而应用在一个最多只需要几千个不同标志符的适当空间里，存储标志符地址对(I, α)的方法。使用这种技巧时，常在表的项目里直接存储标志符I的表达式以及它连接的地址α。这类技巧所依赖的方法称为散列编址。在大规模的、更为实际的情况下，这个方法有重要的实际意义。这类技巧将在第四章中作更深入的研究。下面这个简单的例子，可以说明这个方法。

我们记录并查找在英语中使用得最多的 7 个字母在文章中

出现的相对频率。这7个最常使用的英文字母及其相对出现频率（取自真实的英语文章）列于表1·1中。

表1·1

字　母	出现频率
E	1231
T	959
A	805
O	794
N	719
I	718
S	659

为了简化上面的讨论，根据在字母表中的顺序位置给它写出下标。例如，E是字母表中第五个字母，用E_5表示，A是字母表中第一个字母，用A_1表示。用一个术语Keys（关键字）来表示L_n形式的实体。这里L是一个字母，n是字母在字母表顺序中的位置。现在，用字母L和表1·1所给的频率F构成如下形式的单元：

Key	Freq	Link
L_n	f	∧

例如，对字母n所构成的单元是：

Key	Freq	Link
N_{14}	719	∧

然后，构造由从0～6序次的7个项目的表T。再添写这个表的项目，以得出单元的连接表，如图1·8所示。